Opensugar Cube Quick Start Guide

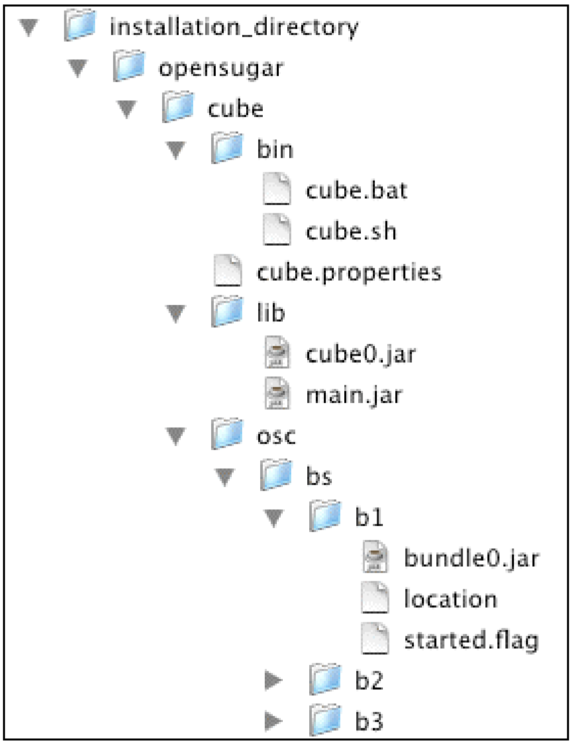
1 Introduction

The Opensugar Cube is Opensugar's implementation of the OSGi Service Platform 2.0 specification. For further information about OSGi and the specification, please refer to http://www.osgi.org.

2 Installation

The Opensugar Cube may be installed on any hardware that supports a Java Virtual Machine. (Java 1.1.8 or above is recommended. Personal Java 1.1.1 may also be used.)

The Opensugar Cube is distributed as a binary installation archive. This file should be extracted to the installation directory of your choice, creating a directory structure similar to that presented in Figure 1.



```
▼ installation_directory
    ▼ opensugar
        ▼ cube
            ▼ bin
                  cube.bat
                  cube.sh
              cube.properties
            ▼ lib
                  cube0.jar
                  main.jar
            ▼ osc
                ▼ bs
                    ▼ b1
                          bundle0.jar
                          location
                          started.flag
                    ▶ b2
                    ▶ b3
```

Figure 1 – Opensugar Cube directory structure

2.1 Start-up scripts

The Opensugar Cube start-up scripts are located in the directory called `opensugar/cube/bin`. Two start-up scripts are normally provided: `cube.sh` for Linux, and `cube.bat` for Win32. These scripts should be edited to reflect the path to your Java Virtual Machine.

2.2 Properties file

Various Opensugar Cube properties are specified in the properties file named `cube.properties` and located in the directory called `opensugar/cube`. This file may be edited to change the cube properties if necessary.

The properties specified in the Opensugar Cube properties file are added to the list of the Java Virtual Machine system properties, when the Opensugar Cube is started. The value of these properties may hence be retrieved (or updated) by bundles as required, using the class `java.lang.System`. The properties file may thus be used as a simple means to specify simple configuration information for the Opensugar Cube, as well as for specific bundles.

Properties specific to the Opensugar Cube are listed in Table 1. In addition to these, the properties file may set properties defined by the OSGi specification. These properties are listed in Table 2. Finally, the properties file may set bundle-specific properties. These should be documented in the bundle-specific documentation.

Property name:	`com.opensugar.cube.libDir`
Description:	Directory in which the Opensugar cube jar and class files are stored
Default value:	“`lib`”
Allowed values:	Any relative or absolute directory path where the required files are located
Property name:	`com.opensugar.cube.classloader.type`
Description:	Classloader type used by Opensugar Cube. Dynamic type may be used in Java 2, allowing the Opensugar Cube framework to be flashed without the need for JVM shutdown. Static type must be used in Java 1.
Default value:	“`dynamic`”
Allowed values:	“`static`”, “`dynamic`”
Property name:	`com.opensugar.cube.stdout`
Description:	Standard output stream. The Opensugar Cube standard output stream may be reassigned to various types of output streams.
Default value:	None (in which case standard JVM output stream is used)
Allowed values:	“`null`” for mute operation, “`file:/path`“ to redirect messages to the specified local file
Property name:	`com.opensugar.cube.stderr`
Description:	Standard error output stream. The Opensugar Cube standard error output stream may be reassigned to various types of output streams.
Default value:	None (in which case standard JVM error output stream is used)
Allowed values:	“`null`” for mute operation, “`file:/path`” to redirect error messages to the specified local file
Property name:	`com.opensugar.cube.stdin`
Description:	Standard input stream. The Opensugar Cube standard input stream may be reassigned to various types of input streams.
Default value:	None (in which case standard JVM input stream is used)
Allowed values:	“`null`” for no input stream, “`file:/path`” to read input from a specified local file
Property name:	`com.opensugar.cube.dir`
Description:	Working directory for Opensugar Cube
Default value:	“`osc/bs`”
Allowed values:	Any relative or absolute directory path that should be used as the working directory
Property name:	`com.opensugar.cube.library.dir`

Description:	Native libraries directory used by Opensugar Cube
Default value:	`“nativeLibs”`
Allowed values:	Any relative or absolute directory path that should be used as the directory where native libraries are stored
Property name:	`com.opensugar.cube.startedFlagFile`
Description:	Flag file that is created by the Opensugar Cube upon successful start-up. If the file is deleted before the Opensugar Cube is started, the presence of the file may be taken as evidence of start-up completion
Default value:	None (in which case the started flag file is not created)
Allowed values:	Any relative or absolute file path that should be used as the cube started flag file
Property name:	`com.opensugar.cube.fudgeSleep`
Description:	An arbitrary pause (specified in milliseconds) that is observed between the restart of bundles on start-up of the Opensugar Cube. This may resolve deadlock situations that may be caused by starting bundles in very quick succession
Default value:	0
Allowed values:	Any non-negative integer
Property name:	`com.opensugar.cube.lowLevelLog.showErrors`
Description:	Flag that indicates whether or not Opensugar Cube error messages should be logged to the standard output stream (these messages do not use the OSGi specified LogService because that service may not always be available.)
Default value:	`true`
Allowed values:	`true`, `false`
Property name:	`com.opensugar.cube.lowLevelLog.showWarnings`
Description:	Flag that indicates whether or not Opensugar Cube warning messages should be logged to the standard output stream (these messages do not use the OSGi specified LogService because that service may not always be available.)
Default value:	`false`
Allowed values:	`true`, `false`
Property name:	`com.opensugar.cube.lowLevelLog.showInfos`
Description:	Flag that indicates whether or not Opensugar Cube information messages should be logged to the standard output stream (these messages do not use the OSGi specified LogService because that service may not always be available.)
Default value:	`false`
Allowed values:	`true`, `false`

Table 1: Properties specific to the Opensugar Cube

Property name:	`org.osgi.service.http.port`
Description:	The port used by the OSGi-specified Http service
Default value:	Implementation-specific. Port 80 is normally used by default
Allowed values:	Any valid port value. Consideration should be given to the required priviledges required on your platform to open a server socket on the specified port

Table 2: Properties defined by the OSGi Service Platform specification

3 Cube start-up

The Opensugar Cube is started simply by navigating to the directory named `opensugar/cube`, and calling the appropriate start-up script:

```
Linux

cd installation_directory/opensugar/cube
./bin/cube.sh
```

```
Win32

cd installation_directory\opensugar\cube
bin\cube
```

> Note: *You **must** be in the directory named* `opensugar/cube` *when the cube start-up script is called!*

After the start-up script has been invoked, some start-up messages may appear at the console. After a few moments the Opensugar Cube is up and running.

> Note: *If the property* `com.opensugar.cube.startedFlagFile` *is defined in the properties file (see Section 2.2), successful start-up of the Opensugar Cube may be tested by checking the presence of the file with the specified name.*

4 Manual bundle manipulation

The most basic method to manipulate bundles is to do it manually.

The Opensugar Cube stores all bundle specific data in the directory named `opensugar/cube/osc/bs`. This directory contains one subdirectory per bundle. The bundle directories are named `b#`, where « # » is the bundle ID.

Each bundle directory *must* contain two files : the bundle jar file, and the bundle location file. The bundle directory *may* also contain other files, such as the « bundle started flag file ».

- The bundle jar file must be named « `bundle#.jar` » where the file index « # » is an integer greater or equal to zero. If several files named bundle#.jar are found in a bundle directory, the jar file with the largest index is used.
- The bundle location file must be named « `location` », and must be a text file containing a single line, specifying the bundle location. The bundle location is a URL that indicates the provenance of the bundle. It is also the URL that is used by default to download bundle updates. *Note that no two installed bundles may share the same location.*
- As its name indicates, the « bundle started flag file » is a flag file that indicates whether or not the bundle should be started when the Opensugar Cube is started. This flag file is named « `started.flag` », and its content is unimportant.

> During start-up, the Opensugar Cube attempts to start all bundles for which a started flag file exits, and leaves other bundles inactive.

Manual bundle manipulation involves the creation / suppression of bundle directories, and the modification of bundle directory contents.

4.1 Bundle install

> Note: *All manual bundle manipulations should be performed while the Opensugar Cube is stopped. The bundle modifications will be taken into account the next time the Opensugar Cube is started.*

To install a new bundle on the Opensugar Cube, a new bundle directory must be created in the bundles directory (`opensugar/cube/osc/bs`). The new bundle directory must be named « b# », where « # » is an integer of your choice (that is not already used).

The bundle jar file must be placed in the new bundle directory, and must be named « `bundle#.jar` », where « # » is another integer of your choice greater or equal to zero. Normally, the name « `bundle0.jar` » is used.

A bundle location file must also be created in the new bundle directory. This file must be named « `location` » and must contain a single line of text in which the bundle provenance URL should be indicated (*e.g.* http://foo.bar.com/bundles/mybundle.jar). A « dummy URL » may be specified in the location file (the effect of this being that bundle updates other than the manual update described below will not be possible). *Note, however, that no two bundles may share the same bundle location.*

4.2 Bundle start

> Note: *All manual bundle manipulations should be performed while the Opensugar Cube is stopped. The bundle modifications will be taken into account the next time the Opensugar Cube is started.*

In order to specify that a bundle should be started on start-up of the Opensugar Cube, a file named « `started.flag` » must be placed in the corresponding bundle directory. The content of the file is irrelevant. Normally, an empty file is used.

> Note: *The presence of the started flag file in a bundle directory is not a gurantee that the bundle will be active after start-up of the Opensugar Cube. It is only an indication that the Opensugar Cube will attempt to start the bundle. Bundle start-up may however be impossible if all package dependencies of the bundle cannot be satisfied.*

4.3 Bundle stop

> Note: *All manual bundle manipulations should be performed while the Opensugar Cube is stopped. The bundle modifications will be taken into account the next time the Opensugar Cube is started.*

In order to specify that a bundle should *not* be started on start-up of the Opensugar Cube, the file named « `started.flag` » in the corresponding bundle directory must be deleted, if present.

4.4 Bundle update

> Note: *All manual bundle manipulations should be performed while the Opensugar Cube is stopped. The bundle modifications will be taken into account the next time the Opensugar Cube is started.*

In order to update a bundle, all jar files in the corresponding bundle directory should be deleted. The updated bundle jar file should then be copied to the bundle directory and named « `bundle0.jar`».

4.5 Bundle uninstall

> Note: *All manual bundle manipulations should be performed while the Opensugar Cube is stopped. The bundle modifications will be taken into account the next time the Opensugar Cube is started.*

In order to uninstall a bundle, the corresponding bundle directory should simply be deleted.

5 The Telnet bundle

The Opensugar Cube is normally provided with a Telnet bundle, which allows local and remote administration of the OSGi framework, using a text-based telnet-type user interface.

> Note: *The Telnet bundle, in its current version, is a tool for developers. It should not be used in hostile environments, as it provides no authentication or access control mechanisms (all incoming telnet connections are accepted).*

The Telnet bundle is a self-contained bundle that may be installed on the Opensugar Cube, or any OSGi 2.0 compliant service platform. When the bundle is started, it creates a server socket on a user-defined port (9198 by default), which accepts incoming Telnet connections. Optionally, the bundle may also start a local console in the shell from which the OSGi framework was launched.

If the Telnet bundle is active on an OSGi framework, and you know the IP address of the framework and the telnet port used, you may open a remote Telnet connection to the framework, with the Telnet client of your choice. If in addition, the local console is enabled, you may also administer the framework locally, using the console provided in the shell from which the framework was launched.
The Telnet and local console user interfaces are identical, and allow full control of the OSGi framework. The set of recognized commands are listed in Table 3.

Command	**Description**
`bundles`	Display list of bundles
`closeconsole` (or `bye` or `quit`)	Close console
`cd url_base`	Set url base that will be prepended to all URL command arguments
`events`	Turn event tracing on/off
`flash url`	Flash cube with download from specified URL (prepended with url_base if any)
`help [command]`	Display list of commands [or specific command help]
`installbundle url`	Download and install bundle from specified

	URL (prepended with url_base if any)
`logs [n]`	Display last n log items (n=10 by default)
`memory`	Display current memory status
`packages [prefix]`	Display list of all packages if no prefix specified, or display package details for all packages with a name that starts with specified prefix
`pwd`	Display current url_base (that will be prepended to all URL command line arguments)
`refreshpackages [id]`	Refresh packages exported by specified bundle (refresh all packages if no bundle id specified)
`reboot`	stop and restart OSGi framework
`run script`	Run specified script (script can be a local file or a URL - in which case it is prepended with url_base if any)
`services`	Display list of services
`shutdown`	Shutdown OSGi framework
`sleep t`	Sleep for t milliseconds
`startbundle id`	Start bundle with specified id
`stopbundle id`	Stop bundle with specified id
`threads`	Display current number of active threads
`uninstallbundle id`	Uninstall bundle with specified id
`updatebundle id [url]`	Update bundle with specified id [download update from specified url (prepended with url_base if any)]

Table 3: Telnet commands

> <u>Note</u>: *Commands are case-sensitive and may be abbreviated (as long as no ambiguity arises).*

Various properties related to the Telnet bundle (including the property that determines whether or not the local console should be enabled) may be specified in the Opensugar Cube properties file. These properties are listed in Table 4.

Property name:	`com.opensugar.service.telnet.port`
Description:	The port on which telnet connections should be accepted
Default value:	`9199`
Allowed values:	Any valid port value. Consideration should be given to the required priviledges required on your platform to open a server socket on the specified port
Property name:	`com.opensugar.service.telnet.console`
Description:	Flag that specifies whether or not a local console should be enabled
Default value:	“`false`”
Allowed values:	“`true`”, “`false`”
Property name:	`com.opensugar.service.telnet.maxConnections`
Description:	The number of simultaneous Telnet connections that should be allowed
Default value:	2
Allowed values:	Any non-negative integer

Table 4: Properties specific to the Telnet bundle